

Una vez realizado el npm init instalaremos las dependencias que necesitamos:

```
Isidro@PC-Isidro:~/Escritorio/DIW_Isidro/T4/Tarea2$ npm install bootstrap @popperjs/core autoprefixer clean-webpack-plugin node-sass nodemon
added 105 packages, and audited 1659 packages in 37s

92 packages are looking for funding
  run `npm fund` for details

93 vulnerabilities (5 low, 30 moderate, 36 high, 22 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
Isidro@PC-Isidro:~/Escritorio/DIW_Isidro/T4/Tarea2$ npm install postcss-cli webpack webpack-cli webpack-merge --save-dev
added 42 packages, and audited 1701 packages in 12s

99 packages are looking for funding
  run `npm fund` for details

93 vulnerabilities (5 low, 30 moderate, 36 high, 22 critical)

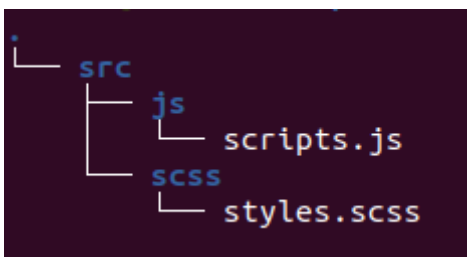
To address issues that do not require attention, run:
  npm audit fix

To address all issues possible (including breaking changes), run:
  npm audit fix --force

Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
```

Creamos la carpeta assets en el directorio raiz y dentro, creamos esta estructura



Dentro de scripts.js y style.scss añadimos las siguientes líneas respectivamente

```
import "../../node_modules/bootstrap/dist/js/bootstrap.bundle.min.js";
@import "../../node_modules/bootstrap/scss/bootstrap";
```

Creamos los 3 siguientes archivos en el directorio raiz del proyecto

```
webpack.dev.js
1 const { merge } = require('webpack-merge');
2 const common = require('./webpack.common.js');
3 module.exports = merge(common, {
4   mode: 'development',
5   watch: true,
6   watchOptions: {
7     ignored: /node_modules/
8   }
9 });

webpack.common.js
4 entry: {
5   scripts: './assets/src/js/scripts.js',
6 },
7 plugins: [
8   new CleanWebpackPlugin(),
9 ],
10 output: {
11   path: path.resolve(__dirname, 'assets/dist/js/'),
12   filename: '[name].js',
13 },
14 };

webpack.prod.js
1 const { merge } = require('webpack-merge');
2 const common = require('./webpack.common.js');
3 module.exports = merge(common, {
4   mode: 'production',
5 });
```

Añadimos los scripts al package.json

```

"scripts": {
  "test": "echo \\\"Error: no test specified\\\" && exit 1",
  "loader": "node-sass --watch scss -o loader/css",
  "grid": "node-sass --watch scss -o grid/css",
  "gallery": "node-sass --watch scss -o gallery/css",
  "genera-doc": "sassdoc scss -d sassdocs",
  "tema": "sassdoc sassdocs --theme vulcan",
  "build": "npm run css && webpack --config webpack.prod.js",
  "css-compile": "node-sass --include-path node_modules --output-style compressed --source-map true --source-map-contents true --precision 6 assets/src/scss -o assets/dist/css",
  "css-prefix": "postcss --replace assets/dist/css/styles.css --use autoprefixer --map",
  "css": "npm run css-compile && npm run css-prefix",
  "watch": "nodemon -e scss -x \\\"npm run css && webpack --config webpack.dev.js\\\""}

```

Hacemos npm run build en la consola

```

isidro@PC-Isidro:~/Escritorio/DIW_Isidro/T4/Tarea2/assets$ npm run build
> tarea2@1.0.0 build
> npm run css && webpack --config webpack.prod.js

> tarea2@1.0.0 css /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2
> npm run css-compile && npm run css-prefix

> tarea2@1.0.0 css-compile /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2
> node-sass --include-path node_modules --output-style compressed --source-map true --source-map-contents true --precision 6 assets/src/scss -o assets/dist/css/

Rendering Complete, saving .css file...
Wrote CSS to /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2/assets/dist/css/styles.css
Wrote Source Map to /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2/assets/dist/css/styles.css.map
Wrote 1 CSS files to /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2/assets/dist/css/

> tarea2@1.0.0 css-prefix /home/isidro/Escritorio/DIW_Isidro/T4/Tarea2
> postcss --replace assets/dist/css/styles.css --use autoprefixer --map

asset scripts.js 76.3 KiB [emitted] [minimized] (name: scripts) 1 related asset
runtime modules 663 bytes 3 modules
cacheable modules 76.4 KiB
  ./assets/src/js/scripts.js 73 bytes [built] [code generated]
  ./node_modules/bootstrap/dist/js/bootstrap.bundle.min.js 76.3 KiB [built] [code generated]
webpack 5.68.0 compiled successfully in 2504 ms

```

Y en la carpeta de assets podremos ver que se ha generado la siguiente estructura

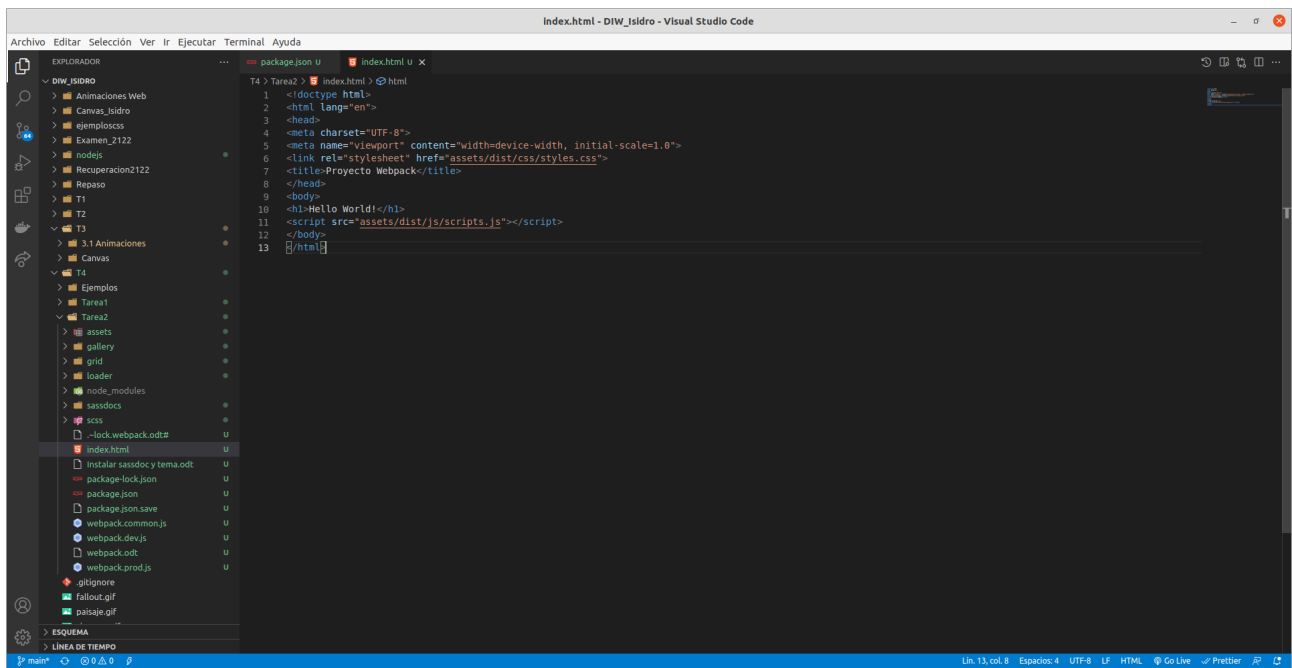
```

.
├── dist
│   ├── css
│   │   ├── styles.css
│   │   └── styles.css.map
│   └── js
│       ├── scripts.js
│       └── scripts.js.LICENSE.txt
└── src
    ├── js
    │   └── scripts.js
    └── scss
        └── styles.scss

6 directories, 6 files

```

Creamos un archivo index.html en la raíz del proyecto donde llamaremos a los dos archivos creados anteriormente(styles.css y scripts.js).



```
1 <!doctype html>
2 <html lang='en'>
3 <head>
4 <meta charset='UTF-8'>
5 <meta name='viewport' content='width=device-width, initial-scale=1.0'>
6 <link rel='stylesheet' href='assets/dist/css/styles.css'>
7 <title>Proyecto Webpack</title>
8 </head>
9 <body>
10 <h1>Hello World!</h1>
11 <script src='assets/dist/js/scripts.js'></script>
12 </body>
13 </html>
```